***Listing of the Claims*:**

1. (Currently Amended) A system for managing software builds, comprising:

a code control system ~~operable to maintain~~ that maintains a code version and a information associated with the code version;

a parser module in communication with the code control system, the parser module ~~operable to parse~~ that parses the information associated with the code version and create a change report, wherein the change report is associated with a state including one of a pending state, an approved state, and a disapproved state; and

a compiler module in communication with the code control system ~~and operable to compile~~ that compiles the code version into an object version based on the change report.

2. (Currently Amended) The system of Claim 1, wherein the information associated with the code version ~~conforms to a standard form and~~ includes comments describing the code version.

3. (Currently Amended) The system of Claim 1, further including a linker module in communication with the code control system, the linker module ~~operable to link~~ links the object version into an executable version.

4. (Currently Amended) The system of Claim 3, wherein the linker ~~is further operable to discriminate~~ determines when the code version is compiled into the object version.

5. (Currently Amended) The system of Claim 1, wherein the parser module ~~is further operable to discriminate~~ determines when the code version has changed.

6. (Cancelled)

7. (Currently Amended) The system of Claim [[6]] 1, wherein the compiler module ~~is further operable to discriminate~~ determines when the change report transitions from the pending state to the approved state.

8. (Currently Amended) ~~The system of Claim 1, further comprising:~~ A system for managing software builds, comprising:

a code control system that maintains a code version and a information associated with the code version;

a parser module in communication with the code control system, the parser module that parses the information associated with the code version and create a change report;

a compiler module in communication with the code control system that compiles the code version into an object version based on the change report; and

a document type definition compiler module that compiles a document type

definition document version into a plurality of metadata class versions and wherein the code control system stores the metadata class versions.

9. (Currently Amended) The system of Claim 8, wherein the metadata class versions are further defined as ~~Vitria BusinessWare~~ metadata class versions of a commercial-off-the-shelf application sold under the trademark VITRIA BUSINESSWARE.

10. (Currently Amended) The system of Claim 1, further comprising an interface definition language grinder module ~~operable to transform~~ that transforms an interface definition language document into the code version.

11. (Currently Amended) The system of Claim 10, wherein the code version of the interface definition language grinder module is further defined as a ~~Java~~ code version of an object oriented programming language sold under the trademark JAVA.

12. (Original) A method of managing software builds, comprising:

changing, by a developer, source code of a software archive maintained by a source archive system;

requesting a build of the software archive including the source code, the request including a build request template;

generating a change matrix based on the build request template;

notifying an approver of the software build request;

notifying the developer when the software build request is denied by the approver; and

rebuilding the software archive maintained by the source archive system based upon the change matrix when the software build request is approved by the approver.

13. (Original) The method of Claim 12, wherein notifying the approver further comprises:

generating a unique number associated with the change matrix; and

providing a test condition

14. (Original) The method of Claim 13, wherein test condition is further defined as a document related to testing the source code.

15. (Original) The method of Claim 12, wherein generating the change matrix includes:

providing a plurality of comments in the build request template;

parsing at least some of the plurality of comments of the build request template into objects; and

providing the objects to change matrix.

16. (Original) The method of Claim 15, wherein the comments include a description, a number associated with the request and information related to changes to the source code.

17. (Original) The method of Claim 16, wherein the number associated with the request is further defined as a unique number.

18. (Original) The method of Claim 16, wherein the changes to the source code include a fields changed portion and an events changed portion.

19. (Original) The method of Claim 12, wherein rebuilding the software archive further comprises:

providing a plurality of data type definition files associated with the software archive;

providing a plurality of interface definition language files associated with the software archive;

manipulating a changed one of the plurality of data type definition files; and

manipulating a changed one of the interface definition language files.

20. (Currently Amended) The method of Claim 12, wherein rebuilding the software archive further comprises:

identifying a plurality of components of the software archive necessary [[for]] to rebuild the software archive;

stamping, with a previous build information, the components of the software archive; and

determining the components to re-compile based on the stamp; and

compiling the components of the software archive based on the stamp.

21. (Original) The method of Claim 20, further comprising compiling only the components of the software archive stamped with a time indicating the component has been modified since a previous build.

22. (Original) The method of Claim 20, wherein the previous build information is further defined as a time stamp associated with each component of the software archive.

23. (Currently Amended) A method for building a software version, comprising:

storing a revised code version and a description of the revisions in a code control system;

generating a change report based on the description of the revisions to the revised code version;

authorizing a build of a software version including the revised code version upon approving the change report; and

building the software version based on the change report upon the authorization of the build of the software version.

24. (Original) The method of Claim 23, wherein building the software version further includes compiling a document type definition document into metadata classes.

25. (Currently Amended) The method of Claim 24, wherein the metadata classes are further defined to be ~~Vitria BusinessWare~~ metadata classes of a commercial-off-the-shelf application sold under the trademark VITRIA BUSINESSWARE.

26. (Currently Amended) The method of Claim 23 wherein building the software further comprises:

grinding an interface definition language document to produce a ~~Java~~ code version of an object oriented programming language sold under the trademark JAVA; and

compiling the ~~Java~~ code version of the object oriented programming language sold under the trademark JAVA.

27. (Original) The method of Claim 23, wherein building the software further includes compiling and linking the code version.

28. (Original) The method of Claim 23, wherein the generating a change report further includes sending an email notification of the change report to one or more administrators.

29. (Original) The method of Claim 23, wherein the authorizing the software build includes sending an email notification of the authorization to one or more software developers.